5К

УДК 004.272.43

З.В. Каляев
НИИ многопроцессорных вычислительных систем им. акад. А.В. Каляева
Южного федерального университета, г. Таганрог, Россия
zakhar@mvs.tsure.ru

Реализация компонентов ОС управления реконфигурируемой вычислительной системой на уровне фрагментов базового модуля

В статье рассматривается реализация операционной системы управления вычислительным ресурсом базового модуля в многозадачном режиме. Данная система позволяет обрабатывать поток масштабируемых параллельных заданий, решаемых на базовом модуле реконфигурируемой вычислительной системы. Приведена структура операционной системы, описаны эксперименты, приведены экранные формы разработанных программ.

Многозадачный режим функционирования многопроцессорных вычислительных систем характеризуется в первую очередь заранее неизвестным потоком входных заданий. Невозможно заранее спланировать, какое количество ресурса будет выделено тому или иному заданию для решения. В связи с этим основным условием существования многозадачного режима для многопроцессорной вычислительной системы является масштабируемость прикладного программного обеспечения.

С целью обеспечения функционирования реконфигурируемых вычислительных систем [1] (РВС) в многозадачном режиме была предложена структура ОС [2], [3], представленная на рис. 1.

В состав ОС должны входить следующие компоненты [4], [5]: подсистема удаленного многопользовательского доступа к базовым модулям (СМД), планировщик заданий, подсистема посттрансляции, загрузчик исполняемых модулей параллельных программ в память вычислительной системы, монитор состояния базовых модулей, система обработки нештатных ситуаций, драйверы и низкоуровневые библиотеки, система тестирования БМ. Каждый из этих компонентов решает отдельную задачу, а их совокупность решает задачу многозадачной ОС РВС.

СМД предназначена для обеспечения обработки удаленных запросов пользователей на использование вычислительного ресурса. Подсистема реализует протоколы прикладного уровня удаленного вызова процедур обращения к вычислительному ресурсу. Подсистема состоит из серверной части, являющейся необходимым компонентом ОС, и клиентской части, исполняемой на машинах-клиентах.

Планировщик заданий предназначен для выделения ресурсов вычислительной системы заданиям из входного потока. Алгоритм планировщика в первую очередь основывается на данных от монитора. Монитор ОС сообщает планировщику текущую карту состояний всех базовых модулей. На основе анализа данной карты планировщик принимает решение о том, сколько базовых модулей и какой задаче выделить для ее решения. Решение также может основываться на приоритете задачи, ее положении во входной очереди, на требовании задачи к минимальному набору ресурсов. Стратегия, алго-

ритм и критерии планирования могут как оптимизироваться, так и модернизироваться в зависимости от области применения вычислительной системы с целью повышения пропускной способности потока задач.



Рисунок 1 – Структурная схема многозадачной ОС РВС

Компонентом ОС, влияющим на решения планировщика заданий, является подсистема обработки нештатных ситуаций. Данный компонент может влиять на работу всех компонентов ОС в случае возникновения нештатной ситуации. Система обработки нештатных ситуаций является автоматом, сопоставляющим действия всех компонентов ОС в случае возникновения нештатной ситуации работы базовых модулей или компонентов ОС. Система имеет как аппаратнозависимые компоненты, которые совершенствуются при модернизации оборудования, так и аппаратнонезависимые компоненты. К нештатным ситуациям может относиться любая ситуация, возникшая в период от начала прихода задания во входную очередь планировщика и до конца ее выполнения, при которой выполнение данного задания или других заданий становится затруднительным или невозможным.

Монитор системы выполняет функцию сканирования состояний базовых модулей (БМ). Монитор способен как в синхронном режиме, так и в асинхронном выдавать информацию о состоянии БМ. Монитор ставит в соответствие времени индексы базовых модулей, занятых решением той или иной задачи. Кроме того, функцией монитора является обеспечение информацией о технической исправности БМ. По запросу от компонентов системы в синхронном или асинхронном режиме в мониторе доступна следующая

информация: номера свободных БМ, номера занятых БМ, номера БМ, занимающихся решением определенной задачи, номера неисправных БМ, информация о состоянии решения каждой задачи.

Драйверы и низкоуровневые библиотеки обеспечивают доступ к каналу БМ, логический доступ к оборудованию БМ, подачу команд и прочее. Библиотеки имеют иерархичное строение и по отдельности могут меняться.

Подсистема посттрансляции исполняемых модулей параллельных программ представляет собой реализацию методов автоматического масштабирования параллельных программ. Входными данными для подсистемы посттрансляции являются задание и массив БМ, выделенные планировщиком заданий для решения [5], [6]. Подсистема выполняет модернизацию исполняемого кода пришедшей программы на то количество вычислительного ресурса, которое выделено для ее исполнения планировщиком заданий. Масштабированный код параллельной программы передается загрузчику параллельных программ.

Загрузчик предназначен для загрузки и инициализации процесса выполнения параллельных программ. Входными данными для загрузчика является исполняемый масштабированный код программы, переданный подсистемой посттрансляции. Загрузчик выполняет заполнение памяти БМ машинным кодом, инициализацию регистров БМ необходимыми значениями.

С целью отладки описанных компонентов ОС РВС были разработаны основные компоненты ОС для управления вычислительным ресурсом, состоящим из одного БМ РВС. Разделение ресурса между заданиями выполнялось на уровне фрагментов (квадрантов) БМ. Необходимо отметить, что при переходе от фрагментов БМ к РВС, состоящей из множества БМ, существенной переработки компонентов ОС не потребуется. БМ РВС состоит из 16 вычислительных ПЛИС, как показано на рис. 2.

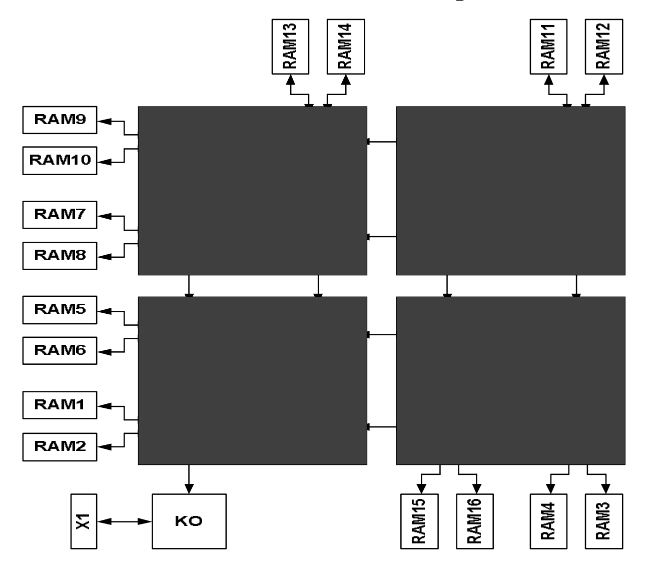


Рисунок 2 – Функциональная схема БМ

Минимальный размер вычислительного ресурса – квадрант – был выбран, исходя из требований решаемых задач к минимальному количеству вычислительного ресурса. Таким образом, вычислительные ПЛИС БМ образуют четыре квадранта, на которых одновременно может решаться не более четырех задач.

Коммутационная система БМ позволяет решать масштабируемые задачи на БМ на любом количестве связанных квадрантов. Каждое из заданий может выполняться на БМ МНМС в соответствии с одной из конфигураций, представленных в табл. 1.

Таблица 1 – Возможные конфигурации выполнения каждого задания на БМ МНМС

	– 0 квадрант		– 1 и 2 квадранты
	– 1 квадрант		– 2 и 3 квадранты
	– 2 квадрант		– 0,1 и 2 квадранты
	– 3 квадрант		– 1,2 и 3 квадранты
	– 0 и 1 квадранты		– 0,1,2 и 3 квадранты

Для отладки разработанных компонентов ОС в терминах масштабируемых параллельных программ [5] были сформулированы три задачи:
– задача фильтрации жидкости в пористой среде;
– задача расчета фильтра с конечной импульсной характеристикой;
– задача умножения матрицы на поток векторов.

Конфигурации трех задач образуют множество сочетаний конфигураций, состоящее из 192 элементов. На рис. 3 представлены некоторые сочетания заданий.



Рисунок 3 – Сочетания конфигураций выполнения заданий

Были проведены эксперименты по обработке потока заданий. Эксперименты с опытным образцом БМ и компонентов ОС РВ представляли собой решение различных потоков заданий на МНМС с разными процедурами планирования.

В общем случае схема запуска задания выглядит так, как показано на рис. 4.

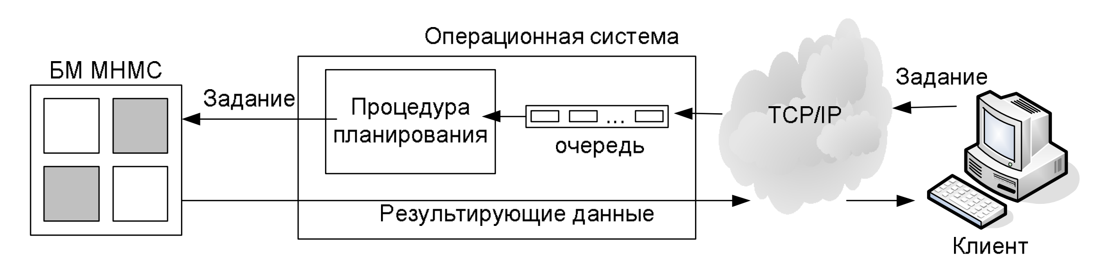


Рисунок 4 – Схема запуска задания

На удаленной машине клиента выполняется соединение с сервером (планировщиком заданий). На машине клиента формируется список файлов, представляющих собой задание. Файлы отправляются на сервер с помощью программы TS.

Сервер при получении задания формирует директорию с уникальным именем и переписывает в нее все до единого файлы задания, полученные от клиента. На сервере создается запись, соответствующая новому заданию, и эта запись помещается в конец очереди заданий.

Файлы задания располагаются в своей директории до тех пор, пока задание не будет в начале очереди и не освободится вычислительный ресурс. Как только это произошло, планировщиком заданий выполняется приостанов выполняющихся в текущий момент заданий, и выполняется перезагрузка конфигурации ПЛИС в соответствии с пришедшим из входной очереди заданием.

После перезагрузки конфигурации выполняемые до приостанова задания продолжают свою работу, производится запуск управляющей программы пришедшего из входной очереди задания. Во время работы управляющая программа задания может быть приостановлена, если будет необходима перезагрузка ПЛИС по причине прихода очередного задания. После своего выполнения управляющая программа создает результирующие файлы, которые передаются обратно клиенту, на клиентской машине может быть выполнена сверка результирующих файлов с некоторым эталоном. На этом задание считается выполненным.

Под потоком заданий понимается совокупность заданий и соответствующее каждому из них время, в которое это задание было отправлено, относительно некоторого начального отсчета времени. Интервалы между посылом заданий могут быть разными, подразумевается, что в условиях многопользовательского доступа к БМ промежутки между приходом заданий на БМ являются случайными. На рис. 5 представлен алгоритм генерации потока заданий.

Zmax – общее количество заданий в потоке;

t – максимальное время между отправкой двух заданий, t характеризует интенсивность заданий в потоке.

Эксперименты проводились для трех разных алгоритмов планирования вычислительного ресурса между заданиями. Первый алгоритм выделяет квадранты БМ МНМС между заданиями в соответствии со следующей формулой:

i = random(I), I – номер секции свободных квадрантов.
j = random(Ji), Ji – количество свободных квадрантов в i-ой секции.

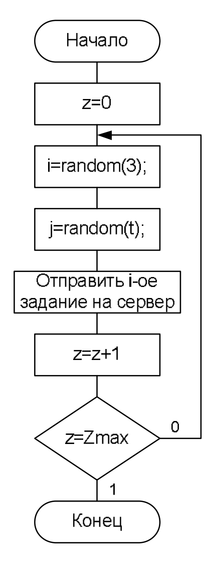


Рисунок 5 – Алгоритм синтеза потока заданий

Для данного алгоритма характерны следующие графики зависимости количества свободных квадрантов от состояния входной очереди (рис. 6). Верхний график соответствует менее плотному потоку заданий.

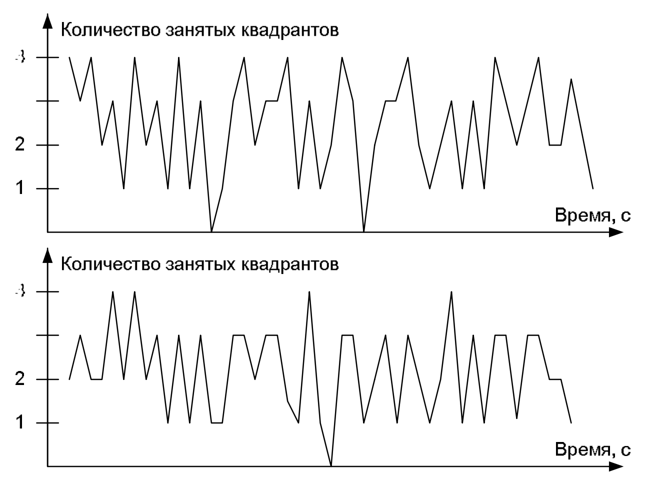


Рисунок 6 – Графики зависимости количества свободных квадрантов от времени

Второй алгоритм выделяет квадранты БМ МНМС между заданиями в соответствии со следующей формулой:

I – общее количество свободных квадрантов;

J – первые задания во входной очереди, J<= I;

n = I/J – квадрантов выделяется J-заданиям из входной очереди.

Для данного алгоритма характерны следующие графики зависимости количества свободных квадрантов от времени (рис. 7).

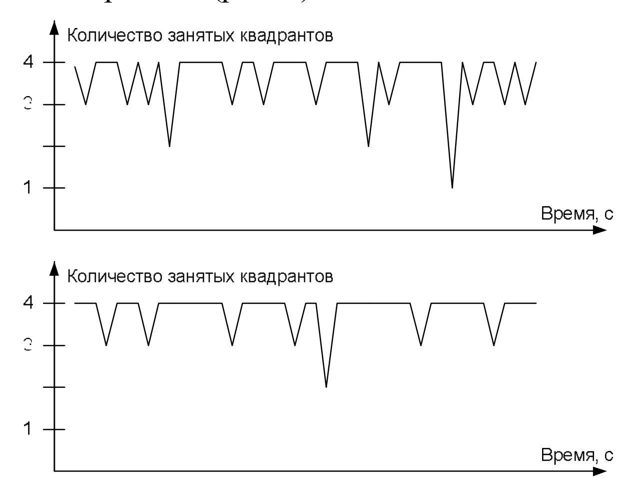


Рисунок 7 – Графики зависимости количества свободных квадрантов от времени

Третий алгоритм планирования выделяет квадранты БМ МНМС между заданиями в соответствии со следующей формулой:

I – общее количество свободных квадрантов;

J – первые задания во входной очереди, J<= I;

n = I/J*Pj – квадрантов выделяется J-заданиям из входной очереди, Pj – приоритет J-го задания.

Для данного алгоритма характерны следующие графики зависимости количества свободных квадрантов от времени (рис. 8).

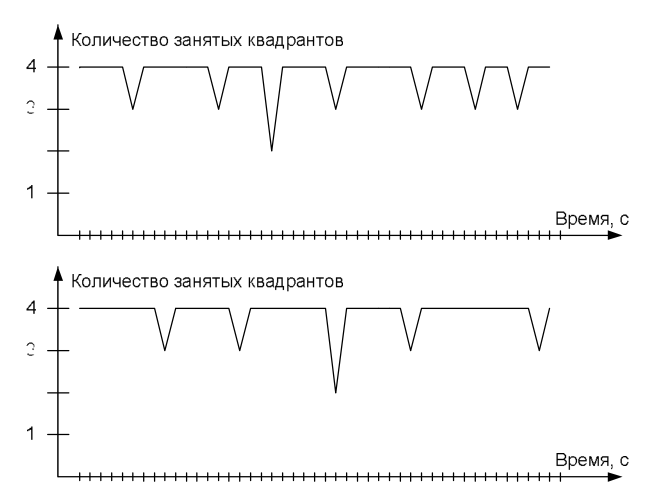


Рисунок 8 – Графики зависимости количества свободных квадрантов от времени

Из графиков видно, что наиболее эффективным алгоритмом распределения ресурса БМ между заданиями для подаваемого потока был последний алгоритм.

На рис. 9 представлена экранная форма планировщика заданий.

На форме отображены квадранты БМ и задания, которые в данный момент выполняются на выделенных им квадрантах. В правом нижнем углу отображаются поля очереди заданий. В правом верхнем углу отображаются запросы пользователей на постановку заданий в очередь планировщика заданий.

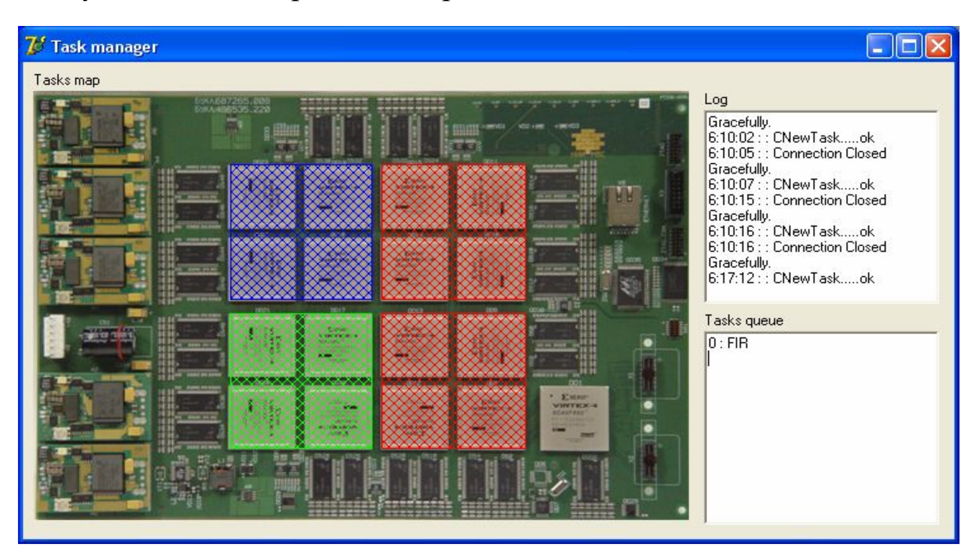


Рисунок 9 – Экранная форма планировщика заданий

На рис. 10 представлена экранная форма программы мониторинга и обработки нештатных ситуаций. Программа анализирует значения токов, напряжений и температуры различных компонентов БМ. В случае превышения критических значений программа примет необходимые действия по обработке нештатных ситуаций.

Реализация основных компонентов ОС для БМ РВС на основе разработанных методов и средств дает возможность использования РВС в эффективном многозадачном режиме с целью минимизации обработки разнородных потоков прикладных задач. Разработанные компоненты ОС могут быть использованы в суперкомпьютерных центрах коллективного доступа, в том числе с возможностью использования вычислительных ресурсов РВС через Интернет.

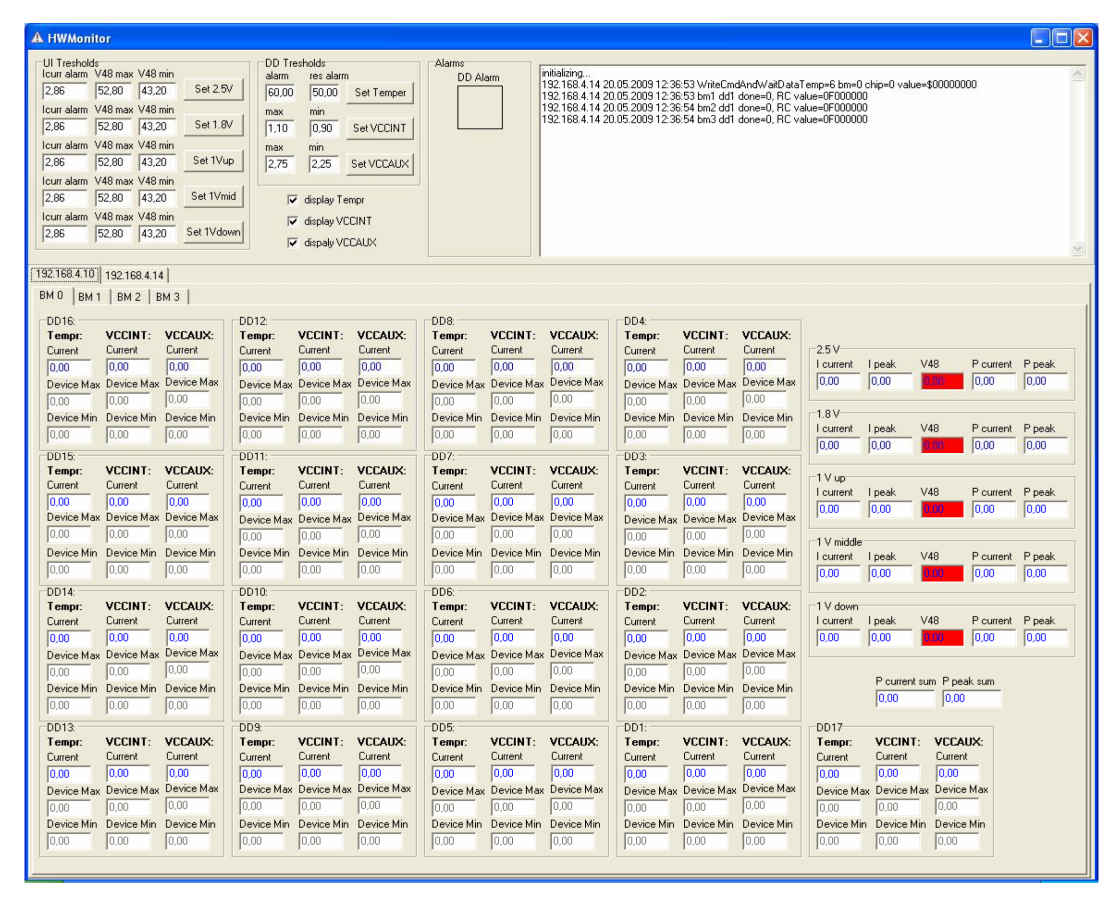


Рисунок 10 – Экранная форма программы мониторинга и обработки нештатных ситуаций

Литература

1. Каляев А.В. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений / А.В. Каляев, И.И. Левин. – М. : Изд-во ООО «Янус-К», 2003. – 380 с.
2. Каляев З.В. Многозадачная распределенная операционная система / З.В. Каляев // Искусственный интеллект. – 2006. – № 3. – С. 144-147.
3. Каляев З.В. Структура многозадачной распределенной операционной системы / З.В. Каляев // Материалы Седьмой Международной научно-технической конференции «Искусственный интеллект. Интеллектуальные и многопроцессорные системы». – Таганрог : Изд-во ТРТУ, 2006. – Т. 2. – С. 102-103.
4. Каляев З.В. Компоненты многозадачной операционной системы для реконфигурируемой вычислительной системы / З.В. Каляев // Материалы Третьей ежегодной научной конференции студентов и аспирантов базовых кафедр ЮНЦ РАН. – Ростов-на-Дону : Изд-во ЮНЦ РАН, 2007. – С. 140-141.
5. Каляев З.В. Система автоматического масштабирования параллельных программ для реконфигурируемых вычислительных систем / З.В. Каляев // Материалы Международной научно-технической конференции «Многопроцессорные вычислительные и управляющие системы – 2007». – Таганрог : Изд-во ТТИ ЮФУ, 2007. – Т.1. – С. 285-289.
6. Каляев З.В. Многозадачная распределенная операционная система многопроцессорной вычислительной системы с программируемой архитектурой / З.В. Каляев, А.Г. Коваленко // Известия ТРТУ. – 2006. – С. 179.

З.В. Каляєв

Реалізація компонентів ОС управління реконфігурованою обчислювальною системою на рівні фрагментів базового модуля

У статті розглядається реалізація операційної системи керування обчислювальним ресурсом базового модуля у багатозадачному режимі. Дана система дозволяє обробляти потік масштабованих паралельних завдань, які вирішуються на базовому модулі реконфігурованої обчислювальної системи. Наведено структуру операційної системи, описано експерименти, наведено екранні форми розроблених програм.

Статья поступила в редакцию 04.07.2009.